

AMENDMENTS TO CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of the Claims:

1. (Currently Amended) **A method of instantiating objects by a virtual machine, said method comprising:**

receiving a first sequence of bytecodes to be executed by said virtual machine;

selecting, at load time, a first-reduced instruction from a reduced set of virtual machine instructions, wherein said first-reduced instruction represents two or more different virtual machine instructions in said first sequences;

translating, at load time, said two or more different virtual machine instructions in said first sequence into said first-reduced instruction from said reduced set of virtual machine instructions;

generating, after said translating, a second sequence of bytecodes that includes said first-reduced instruction, thereby representing said first sequence of bytecodes with a second sequence which includes at least one instruction from said reduced set of virtual machine instruction that replaces said two or more different virtual machine instructions in said first sequence;

determining, at load time, whether said second sequence of bytecodes includes an instantiation instruction immediately followed by a duplicate stack instruction;

generating, at load time, a macro instruction that represents said instantiation instruction and said duplicate stack instruction that immediately follows said instantiation instruction;

loading in said virtual machine prior to execution time, said macro instruction instead of said instantiation instruction and said duplicate stack instruction; and

executing said macro instruction to instantiate a new object.

In a Java computing environment, a Java macro instruction representing:

~~a sequence of Java Bytecode instructions consisting of a Java instantiation Bytecode instruction immediately followed by a Java Duplicate the stack Bytecode instruction;~~

~~wherein said Java macro instruction can be executed by a Java virtual machine operating in said Java computing environment, and~~

~~wherein, when said Java macro instruction is executed, the operations that are performed by said conventional sequence of Java Bytecode instructions are performed.~~

2. (Cancelled)

3. **(Currently Amended)** A Java macro instruction method as recited in claim 1, wherein said Java macro instruction is generated during a the Java Bbytecode verification phase.

4. **(Currently Amended)** A method Java macro instruction as recited in claim 1, wherein said Java virtual machine internally represents Java-instructions as a pair of streams.

5. **(Currently Amended)** A method Java macro instruction as recited in claim 4, wherein said pair of streams includes a code stream and a data stream, wherein said code stream is suitable for containing a code portion of said Java macro instruction, and wherein said data stream is suitable for containing a data portion of said Java macro instruction.

6. **(Currently Amended)** A method Java macro instruction as recited in claim 5, wherein said Java macro instruction is generated only when said virtual machine determines that said Java macro instruction should be generated replace said conventional sequence.

7. **(Currently Amended)** A method Java macro instruction as recited in claim 6, wherein said determination of whether said macro instruction should be generated is made based on a predetermined criteria.

8. (Currently Amended) A method ~~Java macro instruction~~ as recited in claim 7, wherein said predetermined criteria is whether an instantiation instruction is immediately followed by a duplicate stack instruction ~~said conventional sequence has been repeated more than a predetermined number of times.~~

9-21 (Cancelled)

22. (New) A computer system for instantiating objects by a virtual machine, wherein said computer system is capable of:

- receiving a first sequence of bytecodes to be executed by said virtual machine;

- selecting, at load time, a first-reduced instruction from a reduced set of virtual machine instructions, wherein said first-reduced instruction represents two or more different virtual machine instructions in said first sequences;

- translating, at load time, said two or more different virtual machine instructions in said first sequence into said first-reduced instruction from said reduced set of virtual machine instructions;

- generating, after said translating, a second sequence of bytecodes that includes said first-reduced instruction, thereby representing said first sequence of bytecodes with a second sequence which includes at least one instruction from said reduced set of virtual machine instruction that replaces said two or more different virtual machine instructions in said first sequence;

- determining, at load time, whether said second sequence of bytecodes includes an instantiation instruction immediately followed by a duplicate stack instruction;

- generating, at load time, a macro instruction that represents said instantiation instruction and said duplicate stack instruction that immediately follows said instantiation instruction;

- loading in said virtual machine prior to execution time, said macro instruction instead of said instantiation instruction and said duplicate stack instruction; and

- executing said macro instruction to instantiate a new object.

23. (New) A computer system as recited in claim 22, wherein said macro instruction is generated during a bytecode verification phase.

24. (New) A computer system as recited in claim 22, wherein said virtual machine internally represents instructions as a pair of streams.

25. (New) A computer system as recited in claim 24,
wherein said pair of streams includes a code stream and a data stream,
wherein said code stream is suitable for containing a code portion of said macro instruction, and
wherein said data stream is suitable for containing a data portion.

26. (New) A computer system as recited in claim 22, wherein said macro instruction is generated only when said virtual machine determines that said macro instruction should be generated.

27. (New) A computer system as recited in claim 26, wherein said determination of whether said macro instruction should be generated is made based on a predetermined criteria.

28. (New) A computer system as recited in claim 27, wherein said predetermined criteria is whether an instantiation instruction is immediately followed by a duplicate stack instruction more than a predetermined number of times.

29. (New) A computer readable medium including computer program code for instantiating objects by a virtual machine, comprising:

computer program code for receiving a first sequence of bytecodes to be executed by said virtual machine;

computer program code for selecting, at load time, a first-reduced instruction from a reduced set of virtual machine instructions, wherein said first-reduced instruction represents two or more different virtual machine instructions in said first sequences;

computer program code for translating, at load time, said two or more different virtual machine instructions in said first sequence into said first-reduced instruction from said reduced set of virtual machine instructions;

computer program code for generating, after said translating, a second sequence of bytecodes that includes said first-reduced instruction, thereby representing said first sequence of bytecodes with a second sequence which includes at least one instruction

from said reduced set of virtual machine instruction that replaces said two or more different virtual machine instructions in said first sequence;

computer program code for determining, at load time, whether said second sequence of bytcodes includes an instantiation instruction immediately followed by a duplicate stack instruction;

computer program code for generating, at load time, a macro instruction that represents said instantiation instruction and said duplicate stack instruction that immediately follows said instantiation instruction;

computer program code for loading in said virtual machine prior to execution time, said macro instruction instead of said instantiation instruction and said duplicate stack instruction; and

computer program code for executing said macro instruction to instantiate a new object.

30. (New) A computer readable medium as recited in claim 29, wherein said macro instruction is generated during a bytecode verification phase.

31. (New) A computer readable medium as recited in claim 29, wherein said virtual machine internally represents instructions as a pair of streams.

32. (New) A computer readable medium as recited in claim 31,
wherein said pair of streams includes a code stream and a data stream,
wherein said code stream is suitable for containing a code portion of said macro instruction, and
wherein said data stream is suitable for containing data.

33. (New) A computer readable medium as recited in claim 29, wherein said macro instruction is generated only when said virtual machine determines that said macro instruction should be generated.

34. (New) A computer readable medium as recited in claim 33, wherein said determination of whether said macro instruction should be generated is made based on a predetermined criteria.

35. (New) A computer readable medium as recited in claim 34, wherein said predetermined criteria is whether an instantiation instruction is immediately followed by a duplicate stack instruction more than a predetermined number of times.